

Sisällys

1. Johdanto.....	2
2. Mallinnusprosessi.....	3
2.1 Referenssit.....	3
2.2 Tyyllittely.....	5
2.2.1 Ghost town -style.....	5
2.2.2 Sarjakuvamaiset muodot.....	7
2.3 Speed modeling.....	8
2.4 Apumuodot.....	10
3. Mallien tarkentaminen.....	11
3.1 Subpatch ja polygonit.....	11
3.2 Kulmat ja subpatch weight map.....	13
4. Teksturointi käyttäen Lightwave työkalua.....	15
4.1 Layers.....	16
4.2 Nodes.....	18
4.3 Procedural maps.....	19
5. Optimointi.....	20
5.1 Polygonimäärän optimointi.....	21
5.2 Mallit ja radiosity.....	22
5.3 Tekstuurien optimointi.....	23
5.2.1 Materiaalit.....	23
5.2.2 Surface baking.....	24
5.2.3 Image maps.....	25
6. Loppupohdinta	28
 Lähteet.....	 30

1. Johdanto

Undo on mainosalalle suuntautunut yritys ja Ghost Town sen itsenäinen luovuuden hedelmä, animaatiosarja, josta opinnäytettä kirjoittaessani valmistellaan teaseria.

Tällä hetkellä vielä Ghost Town -nimikkeellä kulkeva sarja sijoittuu western-henkiseen hiekan peittämään ympäristöön ja koostuu pääosin monen talon koostamasta kylästä. Koska kyse on sarjasta, kameran on pystyttävä kulkemaan ympäri kylää. Näin ollen mallit on rakennettava kestämaan ympäriajoa, kuten esimerkiksi pelien tasosuunnittelussa otetaan huomioon. Nykyään tietokoneet pystyvät käsittelemään suuria määriä polygoneja laskennassa, mutta määrä ei ole tälläkään hetkellä rajaton.

Ghost Townille ominaista on sarjakuville ja piirrosanimaatioille tuttu muotojen tyyllinen taivuttelu ja rikkominen. Tätä creative director Iiro Harra monesti minulle painotti. Tämä lähestymistapa itsessään määrää optimoinninkin kannalta mallinnustapoja, sillä jotta malleissa esiintyisi paljon kaaria, vaatii se myös polygoneja kurvikkaisiin pintoihin. Tässä kohtaa mallinnustapa eroaa esimerkiksi pelimallinnuksilta, joissa mallien määrä on suuri, mutta polygonimäärät huomattavasti helpompi pitää kurissa, suorien pintojen ansiosta. Optimoinnilla 3d alalla tarkoitetaan tuotosten tarkkuutta suhteutettuna laskenta-aikaan.

Koska kyseessä on kokonainen kaupunki ja animaatiosarja, kaikki siellä esiintyvät esineet on syytä olla visuaalisesti laadukkaita, mutta myös tarpeellisen kevyitä renderointiin. Nämä kaikki lähtökohdat vievät minua mallinnuskeinoissa tiettyyn suuntaan. Luovuuden ylläpitämiseksi on kuitenkin hyvä myös ajoittain ajatuksellisesti irtautua kaavoista ja miettiä vaihtoehtoisia mallinnuskeinoja. Jos mallinnusprosessissa herää ideoita, on tärkeää kirjoittaa ne ylös, sillä ihmismielelle on ominaista unohtaa hetkellisiä oivalluksia.

Osani Ghost Town mallinnuksissa oli vaihtelevaa, sillä suurimmalta osin olin kiinni muissa projekteissa ja kyseisen ympäristön mallinnuksiin osallistuin, kun oli aikaa. Mallinnuskohteet vaihtelivat laidasta laitaan sen mukaan mitä tarvittiin, ulkoympäristöstä sisätilojen esineisiin ja muutaman eri hahmon ensimmäisiin versioihin. Mallinnusprosessin kulku oli tehdä varma suunnitelma ja toteuttaa nopeasti.

2. Mallinnusprosessi

Mallintaminen alkaa lopputuloksen tiedostamisesta sekä prosessin sisäistämisestä. Se on syytä tehdä ajatuksella alkuun ja matkan varrella. Luovuus tulee erityisesti esiin mallinnustavoissa ja lähestymistavoista lopputuloksen luomiseen sekä teknisesti laskennan kannalta kevyen mallin tekemisessä. Aina työtä ei sitäkään ole syytä aloittaa tyhjältä pöydältä. Minun tehtäväksi annetut tietyt Ghost Town rakennukset esimerkiksi pääosin työstin valmiiden jo ympäristöön mallinnettujen rakennusten palasien pohjalta. Monesti joku muu tiimin jäsen oli jo työstänyt versioita.

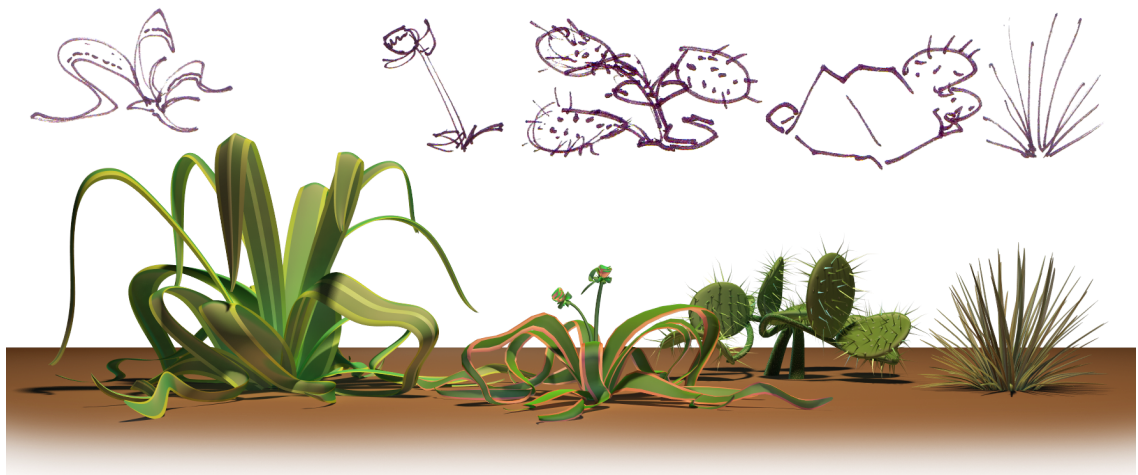
Pixar elokuvan Incredibles tiimissä työskennelleen Neil Blevinsin mallinnustyö on lähellä omaa tehtävääni Ghost Town projektissa. Kuten Blevins haastattelussa kertoo, ajankäytöllä valmisteluun ja tutkimiseen, kuten esimerkiksi materiaalien reagointi valon kanssa ja kuinka esineet todella liikkuvat, on huomattavan tärkeä osa uskottavan ympäristön luomisessa (Blevins).

Taiteellisesti mallinnuksien tavoite ja lopputulos harvemmin ovat itse mallintajan keksintöjä. Tässäkin tapauksessa minun taiteellisia päätöksiäni ohjasivat muutkin Ghost Townia työstävät henkilöt. Kuten Blevins haastattelussa väittää, väitän myös että tietynlaisen tyylitellyn ympäristön luominen on yhtä haastavaa kuin photo-realismia tavoitellessa. Vertaatta työitäsi sitten oikeaan maailmaan, tai elokuvan ja sarjan määriteltyn tyyliin, joudut silti työstämään malleja kunnes ne ovat yhteneviä ympäröivään maailmaan.

2.1 Referenssit

Monesti lopputulos havainnoidaan monien eri kuvien avulla reaali maailman asioista, joita löytyy esimerkiksi Google-kuvahausta. On huomattavasti helpompaa esittää mitä tarkoittaa kuvan avulla kuin tekstillä kuvattuna. Joskus pelkkä suullinen kuvailukin voi johtaa harhaan. Tästä syystä suuntaa antavat valokuvat nekin ovat hyödyllisiä, vaikka eivät vastaisikaan lopullista mallia. Olen huomannut että yksittäisen kuvan voimakkuus on hyvin johdatteleva. Jos taiteellista ohjeistusta antavana ei tahdo sitoa työn lopputulosta liikaa kuvattuun kohteeseen, vaan taiteelliseen visioon, on hyvä viitata moneen eri referenssiin.

Yleisemmin hahmoihin käytetty piirroshahmotelma on suureksi hyödyksi myös ympäristössä. Orgaaniset muodot ja rakennuksetkin monesti on mahdollista luoda ilman hahmotelmia, varsinkin jos kyseessä on geneerinen malli. Mutta jos tavoitellaan tiettyä tyyliä, tai jos ympäristölle on oleellista kiehtova rakenne, on syytä hakea rikkautta ympäristöön kynän, paperin ja visualisoimisen voimin. Materiaaleihin ja muotoihin on huomattavan suuri hyöty valokuvista.



Kuva 1. Referenssit

Kuvan yksi referenssit olivat omia suunnitelmiani, mutta yleisemmin referenssit tulevat art directorin suunnalta, kuten esimerkiksi referenssit hahmomallinnuksiini.

Referenssejä on tärkeää lukea hetken rauhassa. Mallintaessa itse prosessin ennalta hahmottaminen on puolet työstä. Referenssit ympäristöön, jos ne eivät ole suoria ohjeita, on hyvä mieltää ohjaavina eikä suinkaan orjallisina kaavoina yhtäläisyyteen mallin ja kuvan välillä.

Suorat ohjeet, tarkat esimerkkikuvat ja värimäärittelyt kuitenkin usein tarkoittavat sitä, että mallin pitää sisältää joitain elementtejä esimerkiksi käsikirjoitukseen sisältyvistä syistä, tai oltava yhtenäinen muuhun jo suunniteltuun. Mallintaessa on hyvä kääntyä art - tai tässä tapauksessa creative directorin puoleen mahdollisimman usein jos jo alkuun haistaa muutoksien mahdollisuuden, jotta vältetään liiallisen työn määrältä. Joskus alkuun hyväksytyt mallit nekin korvataan joka tapauksessa paremmin toimivilla uusilla malleilla. Mallintajan on hyväksyttävä tämä tekijä jo alkuun, sillä harva asia säilyy muuttumattomana.

2.2 Tyylittely

Yleensä sarjakuvamaiseksi kuvatun tyylin luomiseen vältetään täysin suoria kulmia sekä täysin suoria sivuja. Tyylittely on helpoin tehdä jälkikäteen, sillä reaali-mittakaavaan mallintaessa polygonien topologia ja pisteiden yhtäläiset vektorietäisyydet rakentuvat varmemmin. Jotta mallit voi muljauttaa lopulliseen sarjakuvamaiseen tai muuhun tahdottuun muotoon, ohjelmat kuten lightwave tarjoaa hyviä etäisyyteen perustuvia muokkaustyökaluja, kuten magnet tool, drag network tool, bend, swirl, taper. Oli mallinnusohjelma mikä tahansa, tietyt taivutus- ja venytys-työkalut löytyvät varmastikin lähes kaikista ohjelmista.

Suosittelavaa on hakea muoto ensin lowpoly-geometriaan apumuodoin ja muovata itse high poly-malli apuleyerin pohjalta. Näin vältetään liiallisilta ja turhilta työkalujen käytöltä ja ei-toivotuilta venymisiltä polygoni-geometriassa. Toinen vaihtoehto on mallintaa suoraan tahdottuun muotoon, mutta tällöin polygonien suhteet toisiinsa, muodot ja kulmat on huomattavasti haasteellisempaa pitää kurissa. Smooth pintojen ja teksturoinnin kannalta on usein olennaista, että yksittäiset polygonit ovat kohtuullisen tasaisia.

2.2.1 Ghost Town style

Ghost Town- ympäristölle ominainen tyyli on kenties hallittu sarjakuvamaisuus. Mahdollisia vertauskuvia muotoihin löytynee Lucky Lucke-sarjakuvista (kuva 2), mutta ympäristöön itsessään myöskin elokuvamaailman klassikko lännen maailmoista, kuten Clint Eastwood henkisistä elokuvista, jotka ovat monille luoneet mielikuvat lännenmaailmasta (kuva 3). Lähtökohtina ovat myöskin esimerkiksi historialliset aidot hahmo portrait kuvat sen ajan kuvaustekniikalla, sekä historian tietämyksellä sävytetyt yksityiskohdat.



Kuva 2. Lucky Luke, alun perin Maurice de Bèrè piirtämä ja kirjoittama sarjakuva. Myöhemmin kirjoittajana toimi pitkään René Goscinny. Tällä hetkellä sarjakuvaa piirtää ranskalainen taiteilija Achdé.



Kuva 3. kuvia elokuvista Hyvät pahat ja rumat, sekä Muutaman dollarin tähden

Nämä esimerkit eivät toki ole Ghost Town itsessään, vaan pelkästään referenssejä antamaan mielikuvia ja kasvattamaan kokonaisuutta. Lopullisesta tyyli ja maailma rakentuu tiimin kesken luovassa ympäristössä. Mikä tekee käsikirjoituksen, hahmot, niiden välisen vuorovaikutuksen ja sitä kautta maailman, on niin suuri tekijä että sitä on syytä heijastaa ympäristöön.

Ghost Town mallinuksissa olen ottanut huomioon hyväksyttämisen, eli mahdolliset muutokset ja mahdollisen uudelleen mallintamisen. Muutoksienkin takia on syytä säilyttää työvaiheista monta versiota. Joitakin tyyllillisesti toimivia osia malleissa voi käyttää vaihtoehtoisesti toisessa mallissa. Paloja malleista on näin hyvä myöskin mallintaa omille layereilleen, ja yhdistellä lopulta versiossa, joka viedään itse scene tiedostoon, eli lopulliseen laskettavaan ympäristöön.

Ghost Town ympäristölle yhteinen tekijä lienee muotojen kaarevuus ja elävyys. Päädyin ratkaisuun tehdä suurimman osan mallinuksista ensin staattisempana ja muovata valmiimpaa mallia sitten pidemmälle ja eläväksi siihen soveltuvilla työkaluilla.

Lopulliset mallinukseni hyväksyi Creative Director Iiro Harra. Varsinainen ympäristön mallintaminen oli aloitettu kauan ennen omaa panostani, joten Ghost Town tyylin kartoitti mallinuksissani suurelta osin myös Tuomas Kankolan valmiit mallit, joita heijastin omiin tuotoksiini.

2.2.2 Sarjakuvamaiset muodot

Mallinnuksiani määritteli myös se, mikä tekee yleisesti animaatioelokuvista ja sarjoista piirrosmaista ja sarjakuvamaistakin. Tekniikan kehityksen myötä johtavat animaatiofirmat ovat luoneet tietä tiettyyn suuntaan, kuten esimerkiksi Pixar, ja toimivat tyyllisten käsityksen luojina siitä mitä on animaatio parhaimmillaan. Polygonien määrän kasvu on mahdollistanut kaiken sen luomisen, mikä oli ennen mahdollista vain oikeilla pienoismalleilla ja nukeilla kuvattuna, sekä viemään tyyliä täysin piirrosmaiselle asteellekin kuten esimerkiksi lyhyt animaatiosarja Usavich. Kolmiulotteisilla malleilla jäljitellään paljon tyyllisiä temppuja, jotka on jo piirrosanimaatioissakin aikoinaan todettu toimiviksi, josta suurimman työn on kenties tehnyt aikoinaan Disney.

Tässä tapauksessa sarjakuvamaisuuteen viitaten puhun muodoista, jotka tuovat persoonallisuutta yksinkertaisiinkin muotoihin ja eloa staattisiin elementteihin. Tyyllisesti sarjakuvamaisuutta kuvaamaan on helpoin liittää liioittelu ja ympäristön lähes karikatyyrimäisyys. Sarjakuville olennaista on muotojen poikkeama reaaliaamailmasta tyylin määrittelemien määrin. Kuten Lucky Luke sarjakuvissakin hahmot sekä ympäristö on tyylieltyä, mutta toistuvalla kaavalla tuotannossa ja hillityin määrin.

Kuten sarjakuvissakin sarjatuotantona ympäristössä ja hahmoissa on oltava toistuvat muodot, jotta pitkällä tähtäimellä tyyli pysyy yhtenäisenä läpi tuotannon. Myös Ghost Townia määrittelee se, mikä on tiimin sisäinen käsitys tyylistä ja mikä tekee eri palasista yhtenäistä. Yleinen näkemys ja yleinen sarjakuvamaisuus tyyllilajina on kuitenkin eriteltävä mielessä siinä määrin Ghost Townista, että kyseessä on kuitenkin itsenäinen tyyllilaji ja sen luominen.

Sarjakuvamaisuus mielikuvana, animaation visualisoiminen käsikirjoituksen pohjalta, ja kaikki se mikä tekee Ghost Townin, oli ensiaskel, josta seuraava vaihe oli mallinnuskeinojen määrittelemine.

2.3 Speed modeling

Speed modeling on harrastajien kesken käytetty termi pikamallinnuksesta eli tehokkaasta mallinnuksesta. Pikamallinnus käsittää laadukkaan työn tekemisen mahdollisimman pienessä ajassa. Nopeus mallintaessa on yhdistelmä harjaannusta ja hahmottamiskykyä. Muotoja on hyvä miettiä mahdollisimman yksinkertaisista rakenteista, sillä polygonimallien työstössä polygoneja on aina helpompi lisätä kuin vähentää. Nopea mallinnusprosessi siirtää työprosessissa enemmän aikaa loppupäähän eli mallien viimeistelyyn.

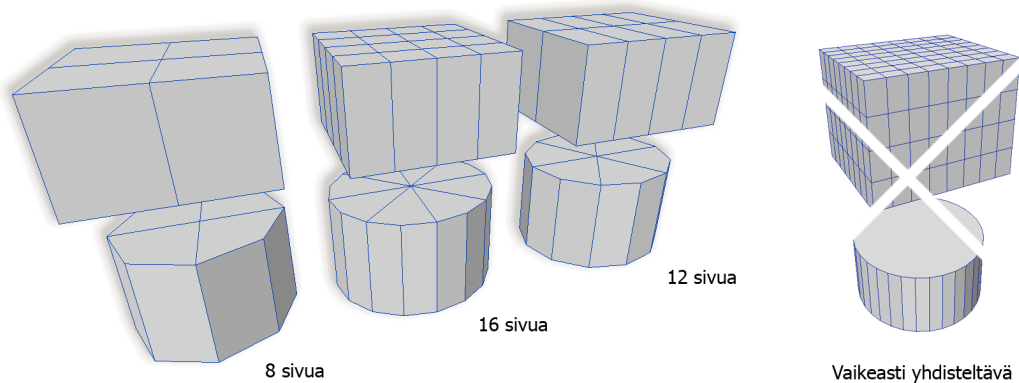
Nopea mallintaja osaa hahmottaa reaalielämän esineistä ja ympäristöstä eri perusmuotoja ja yhdistelemään niitä toisiinsa, joista kokonaisuudet muodostuvat. Useimmiten ympäristön voi pelkistää hyvin yksinkertaisiin muotoihin.



Kuva 4. Ghost Town revolveri

Kuvassa neljä on esimerkillinen revolveri, jonka työstin loppuun, aloittaen hyvin yksinkertaisesta geometriasta, tarkentaen mallia tasaisesti, tarpeen mukaan. Revolverissa on hyödynnetty eri muotojen eli tarkemmin polygoniryhmien yhdistelemistä ja näin ollen mallia voi tarkentaa subdivisionia hyödyntäen huomattavasti myös lähikuviin.

Polygoniryhmiä on syytä kuitenkin yhdistellä vain tarvittaessa, sillä polygonimäärät kasvavat huomattavasti käytettäessä mikrodetaileja eli paikoittain yksityiskohtaisia malleja. Jos katsoja ei koskaan näe yhdistettyä kohtaa, se on turha. Tässä tapauksessa mallin lähitarkastelu on kuitenkin ennen pitkää todennäköistä.



Kuva 5. Ympyrän polygonimäärässä on hyvä käyttää parillista jaettavaa lukumäärää, kuten 8,12,16 sivua.

Mallit olisi syytä rakentaa pääosin alkuun kauttaaltaan samoissa suhteissa polygoneja, jotta subdivisionia käytettäessä malli ei tarkennu epätasaisesti, jolloin taas tarvittaisiin mahdollisesti suurempia arvoja tarkentamaan yksittäisiä osia mallista. Myöskin osien yhdisteleminen tarvittaessa on huomattavasti helpompaa, kun polygonien määrä on alkuun pieni. 8-sivuinen ympyrä tosin jo joko vaatii subdivisionin tai manuaalisen polygonimäärän kasvattamisen. Subdivision pehmennyksen käyttäminen ei aina ole järkevää.

Mallien tarkentaminen on useimmiten hyvä tehdä mallinnuksessa viimeiseksi, kun perusmuodot ja mallin topologia lähestyy oikeaa lopputulosta, mutta kuitenkin ennen teksturointivaihetta. Tasapainottelu subdivisionin ja jakamattoman polygonin välillä, vaihtelee mallin mukaan.

Topologialla tarkoitetaan polygonien järjestystä ja niin kutsuttujen looppien eli polygonijatkumoiden kiertämistä mallin sisällä, polygonien hierarkiaa subdivide jakamisen ehdoin. Topologia liitetään usein kasvomallintamiseen ja kasvomallinnus onkin tehokkain esimerkki kuvaamaan polygonien järjestyksen merkitystä. Topologinen ajattelu on hyvä pitää mielessä kuitenkin aina, kun hyödyntää subdivision jakoa.

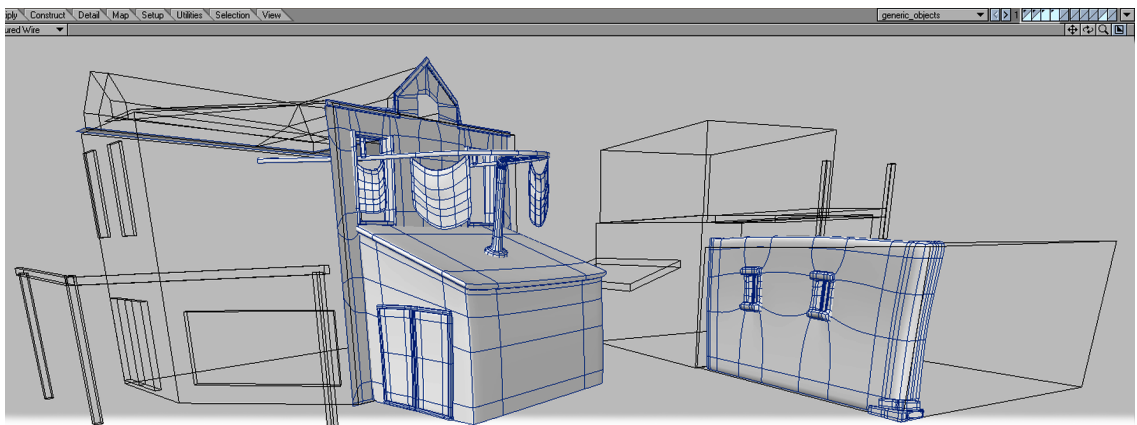
Ottaen huomioon mallien suuren määrän ympäristössä, rakennuksissa subdivisionin käytössä ja sitä myöten polygonien määrässä voi säästää kulmissa käyttäen listoja, lautoja sekä muita esineitä peittämään kulmakohtia. Näin ollen kulmiin ei tarvitse aina

mallista riippuen lisätä polygonimäärää. Kulmien pehmentäminen jollakin keinolla on useimmiten mallien kannalta vaatimus, sillä jos katsomme omaa ympäristöämme, harvemmin näemme missään täysin teräviä, ikään kuin teollisen mittakuvan kulmia.

Joskus normal mapin käyttö sekin erityisesti tasaisten pintojen keskellä on tehokas ratkaisu luomaan illuusiota elävästä pinnasta ja vähemmän teollisesta pinnasta. Normal map teksturointia hyödynnetään paljon varsinkin pelien kenttäsuunnitteluissa. Animaation ympäristöönkin se on tehokas väline niin kauan kuin normal mapit ovat yhteneväisiä yleiseen tyyliin. Mallinnusten teksturoinnin nopeuttamiseksi mainittakoon myös Eetu Martolan grit node, jota käytin mm. kivissä teksturoimaan syvennyksiä, sekä kuvassa 19 esiintyvässä kolikossa. Jotta välttää ylimääräisen työn tekemisen, teksturointiakin on syytä miettiä läpi mallinnusprosessin.

2.4 Apumuodot

Suuria malleja kuten rakennuksia mallintaessa on hyödyllistä referenssien lisäksi käyttää raakamallinnuksia, eli yksinkertaistettuja malleja havainnoimaan muotoja. Siinä missä kynä on piirtäjälle työväline keskusteluun paperin ja ajatuksen välille, yksinkertaiset muodot auttavat mallintajaa hahmottamaan jo ennalta polygonirakennetta, sommitelmia ja alueita mihin tarvitaan yksityiskohtia.



Kuva 6. Apumuodot omalla tasollaan

Yksi tehokas tapa mallintaa, on keskittyä yksityiskohtaisesti jatkuvasti malliin pitämällä raakamallista haamukuvaa aktiivisen tason alla. Apumuodoilla voi selvittää sijainnit placeholderina eli paikallisina myöhemmin korvattavina muotoina ja selvittää suhteet muihin mallinnuksiin itse scene tiedostossa. Lightwave sisältää myös toimintoja, jotka

käyttävät tasoja hyödyksi kuten esimerkiksi boolean. Boolean tosin automatisoiduista työkaluista on hyvin epäluotettava, ja itse olen käyttänyt vain silloin kun polygonirakenne, kuten vaikka puulautaan kaiverrettu teksti on niin työläs tehdä, että aika ei riitä käsityöhön. Array työkalut taas ovat suuri hyöty ja ne on syytä opetella heti alkuun sillä rutiinit herkästi vievät mallinnuksen turhankin manuaaliselle tasolle.

3. Mallien tarkentaminen

Ympäristöä mallintaessa isompaa animaatioprojektia varten, on otettava huomioon mallien uudelleenkäyttö eri kuvakulmista. Mallien geometria on syytä olla tarkennettavissa, kun esineitä ja ympäristöä kameran etäisyydestä riippuen eri leikkauksissa mahdollisesti tuodaan esiin. Vaihtoehtoisia lähestymistapoja yksityiskohtien luomiseen on monia. Yksi tehokas keino rakennuksissa optimoinnin kannaltakin on tehdä erillisiä kulmaosia kuten lautoja, jolloin pyöristysä ja tarkennuksia ei tarvita alkuperäisen mallin kulmiin. Teksturoinnissa uv mappauksen kannalta tasaiset yksittäiset talon sivut on helpompi määrittää ilman yhteneviä kulmia.

3.1 Subpatch ja polygonit

Lightwavessa mallinnus ja animointi tehdään eri ohjelmilla. Eri ohjelmien käyttäminen mahdollistaa mallien päivittämisen lopullisiin ympäristöihin eri tahoilta samanaikaisesti ja versioiden säilyttämisen jokaisesta itsenäisestä mallinnuksesta erikseen. Malleista voi myöskin tuoda sceneen paloittain layerit, mikä on ympäristössä liikkuviin mekaniikkoihin rigauksen kannalta olennaista, ja mahdollistaa palasien liikuttelun vielä scenessä. Animaation kannalta on mahdollista tuoda sisään animaation data toisesta täysin eri ohjelmasta ja näin jakaa työskentelyn monen eri tiedoston ja ihmisen välillä.

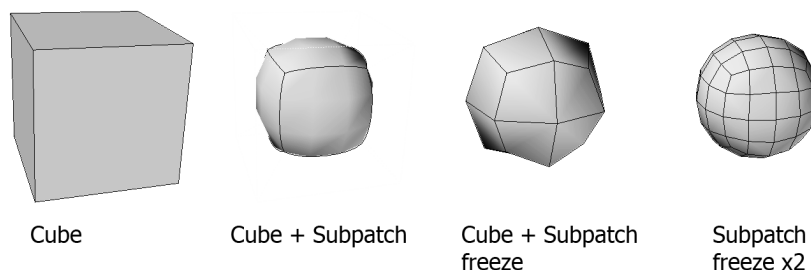
Subpatch on lightwave-ohjelmassa toinen ohjelman käyttämistä polygonimäärää kasvattavista moodeista. Vastaavaa tekniikkaa kulkee esimerkiksi 3dStudio Maxissa turbosmooth ja meshsmooth nimikkeillä. Toinen moodi on Catmull Clark, joka mahdollistaa yli neljäsiivuisten polygonien eli ngons:ien jakamisen. Catmull Clark on näistä raskaampi, mutta sopii kenties tarkkaan malliin paremmin, kuten esimerkiksi auton malliin (Newtek forum). Suuret ympäristöt suosivat nopeudessa Subpatch jakajaa. Koska useimmat ohjelmat jakavat parhaiten nelikulmaisia polygoneja, yhteensopivuus syistä en suosittele ngonsien käyttöä.

Jos ympäristö on luotu quadeista eli neljäisivuista polygoneista, malleja voi tarkentaa, eli moninkertaistaa polygonimäärän, subpatchin arvoa kasvattamalla. Subpatch arvoa kasvattamalla polygonimäärä tuplaantuu kertoimin scene tiedostossa eli Lightwave renderointi ja animointi ohjelmassa malli ja layer kohtaisesti.

Kaikkea ei ole todellakaan syytä subpatch pehmentää ja jakaa. Vaan optimoinninkin kannalta isoihin ympäristöihin, on syytä lisätä polygoneja vain tarvittava määrä ja tarvittuihin paikkoihin, joissa polygonit näkyvät läpi lopullisissa kuvissa. Esimerkiksi Ghost Town rakennuksissa ei ole syytä mennä lähikuviin kulmista, ja jos on, on se niin erikoistilanne, että sitä varten voi teettää omat mallit. Yleensä tällöinkin kuvakulmasta johtuen kulma peittyy depth of field pehmennykseen kohdistuksen ollessa kamerassa taka-alalla.

Pääasiassa kuitenkin mallinnuksiin pätee se yleinen sääntö että polygonit eivät saa näkyä ja varsinaisia täysin veitsen teräviä kulmiakaan ei ympäristössä ole olemassa. Reunat usein vaativat näin jonkin asteisen pehmennyksen. Lopputulos on tasapainottelua optimoinnin ja tarkkuuden välillä. Kulmien pehmentäminen ei tarvitse aina olla geometrista. Teräviä kulmia voi pehmentää esimerkiksi myös tekstuurilla.

Joskus mallin voi myöskin jakaa manuaalisesti, shift + d, ja esimerkiksi band glue toiminnolla tämän jälkeen yhdistää tarpeettomia polygoni looppeja tasaisilta alueilta. Yksi tapa mallintaa on luoda pallomaiset muodotkin pehmentämällä esimerkiksi neliön subdivauksella ja hyödyntää samaa lähestymistapaa eri muotoihin.



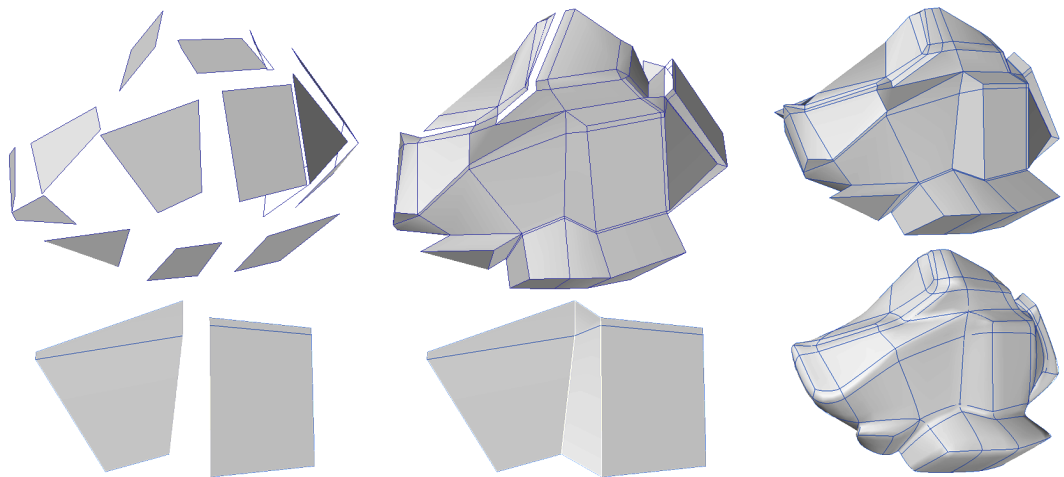
Kuva 7. Subpatch freeze

3.2 Kulmat ja subpatch weight map

Jos on ennakoitavissa että malleja tai ympäristöä tullaan näyttämään lähietäisyydeltä, on hyvä jo mallinnusvaiheessa kasvattaa polygonimäärää kulmiin tarvittava määrä. Tällöin subpatchia käyttäessäkään kulmat eivät pyöristy ei-toivotulla tavalla.

Kulmia on mahdollista myöskin kiristää subpatch weight mappia hyödyntäen. Ghost Town ympäristöön tarvittiin kiviä ja kallioita. Mietin mallinnustapaa eroosion eli tuulen kuljettaman hiekan muokkaamien sekä terävien että pyöreiden, kiehtovien muotojen avulla.

Jotta sain suuren määrän erilaisia kallioita ja kiviä tuotettua, päädyin mallintamaan tasaisia pintoja, polygonitasoja lähekkäin, ja sitten yhdistelemällä ne toisiinsa hyvinkin karkeina malleina. Jotta kiville ominainen terävyys ja sitä kautta massankin tuntu tulisi esiin, kulmiin oli jätettävä terävyyttä. Koska kivet eivät luonnossakaan esiinny kaikki yhtäläisinä, vaan muodot ovat riippuvaisia monista tekijöistä ja näyttävät enemmän satunnaisilta kuin suunnitelluilta, tein polygoneja suurimmalta osin satunnaisesti, mutta kuitenkin pitäen mielessä lopullisen mahdollisen polygoni geometrian.

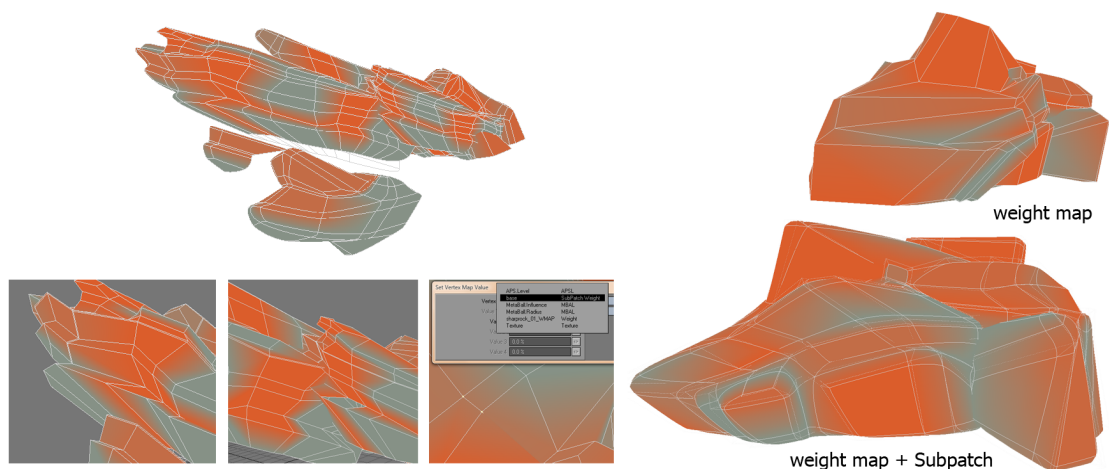


Kuva 8. Polygonien yhdisteleminen

Polygonien yhdistelemisen kannalta on hyvä katsoa, että yhdisteltävät osat ovat toistensa kanssa polygonimäärältään yhtäläisiä. Kuten kuvassa 8, geometria on alkuun hyvin yksinkertaista. Yksinkertaisen geometrian jakaminen subpatchin avulla tekee pyöristettyjä muotoja. Lopputulos vielä vaati terävyyttä.

Nopea looginen ratkaisu terävöittämiseen oli subpatch weight mapin hyödyntäminen. Weight mapit ovat vertex karttoja Lightwavessa joita voi hyödyntää mm. deformaatioon, hahmon rigiin ja väritykseen. Värityksessä vertex mapeista on suuri hyöty esimerkiksi alpha kanavien määrittämiseen päällekkäisille tasoille. Christopher Evans on esimerkiksi tehnyt weight mapin käytöstä lyhyehkön tutoriaalin (Evans 2003). Subpatch weight map löytyy Lightwavessa weight listasta nimikkeellä SubPatch Weight, joten sitä ei tarvitse luoda erikseen.

Jotta verteksien määrittely olisi mahdollisimman nopeaa, ja jotta subpatch weightien käytöstä hyötyy, mallin on tähän tarkoitukseen hyvä olla yksinkertainen eli luotu vähäisellä polygonimäärällä. Arvot Lightwavessa valitaan valikosta map, set map value, tahdottu arvo sekä valiten subpatch weight listasta. Tai vaihtoehtoisesti weight tool map-valikosta, joka mahdollistaa numeric toolin, pikanappi n, ja näin falloff valintojen tekemisen. 100% arvo vastaa täysin pehmentämätöntä kohtaa ja 0% pyöristää mallia normaalisti eli muokkaa surface tasojen poikkeamaa sekä liukumaa näihin vertex pisteisiin nähtynä subpatchille ominaiseen tapaan.



Kuva 9. Subpatch Weight map

Huomioitavaa on kivistä esiintyi weight mapista alkuun ongelma, sillä olin käyttänyt terävimmissä kohdissa 100% arvoja, jolloin polygoneja ei tietenkään paikoittain voi koskaan pehmentää, vaikka subpatch jakoja kasvattaisi äärettömyyksiin, ja näin ollen lähikuvien käyttäminen on mahdotonta. 75% oli korkein käyttökelpoinen arvo, jolloin pyöristykset kasvavat vielä kulmiin ja kiristävät kulmia, kuitenkin vielä luoden mikrodetailitasolla pyöristyksiä. Weight mappiin voi vielä jälkikäteen määrittää korkeimman ja alhaisimman mahdollisen arvon jälkikäteen, joten korjaus kaikkiin malleihin ei ollut työläs prosessi.

0% weight map arvojen käyttö on ehdotonta tässä tapauksessa, sillä mallin alimman arvon kasvattaminen hävittää koko subpatchin tarkoituksen. Jos mallissa alin arvo esimerkiksi on 20%, tasaisilla pinnoillakin polygonimäärä kasvaa mutta ulkoisia muutoksia ei näy, jolloin polygonimäärä kasvaa turhaan jakoja lisätessä lopullisessa scene tiedostossa.



Kuva 10. Kivet renderoituna

Weight mappeja voi käyttää rajattoman määrän. Kiviin tein myös toisen weight mapin kertomaan teksturoinnissa minne alueille miksataan maan tekstuuria eli hiekkaa.

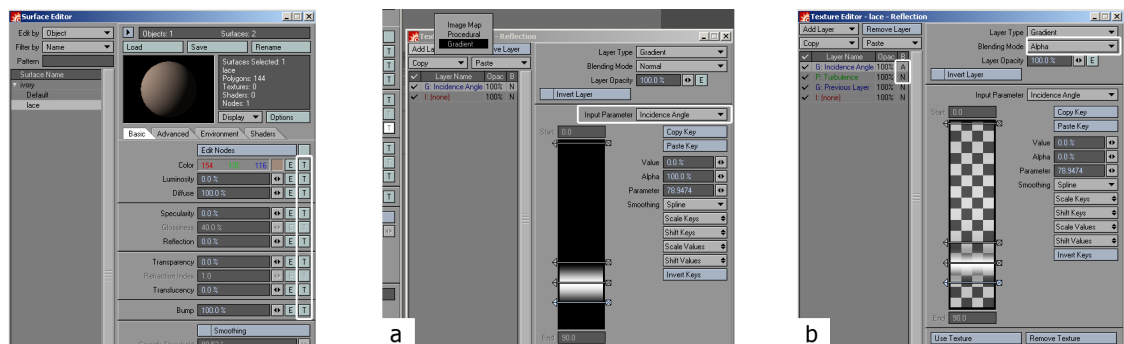
4. Teksturointi käyttäen lightwave työkalua

Teksturointi Lightwave ohjelmassa suoritetaan surface editorin kautta joko ikkunan valmiita kanavia käyttäen, tai edit nodes pohjasta. Nodeista puhuessa viitataan node puuhun kuvastaen nodeille ominaista monihaaraisuutta. Teksturointi on huomattavasti hyödyllisempää rakentaa jo alkuun node pohjaisena, sillä siirryttäessä node pohjaiseen ympäristöön, jo säädettyjä kanavien yhdistäminen node puuhun ei varsinaisesti onnistu. Pienet säädöt ja kokeilut on tosin huomattavasti nopeampi tehdä surface editorin ensimmäiseen ikkunaan, ja ikkunan texture editor alavalinnoista voi luoda toimivia tekstuureja gradient, procedural ja image map layereitä hyödyntäen. Monesti tekstuureja säädetään jälkikäteen myös muidenkin toimesta, jonka takiakin nodet ovat selkeämpi lähestymistapa, sillä node rakenteesta näkee nopeasti kaikki toiminnot avattuina, kun taas surface editorin ikkunaan tehdyt säädöt on kaikki avattava erikseen. Texture editor sopii näin ollen hyvin yksinkertaisiin tekstuureihin ja mitä monimutkaisempaa tekstuuria luo sitä tärkeämmäksi tulee käyttää node pohjaa.

Jos on täysin uusi Lightwave käyttäjä, on erittäin tärkeää huomata, että objektit on tallennettava layoutissa tehtyjen tekstuurimäärittelyjen säilyttämiseksi. Kaikki muutokset häviävät ilman save object toimintoja. Tämä on lightwave ohjelmarakenteen rasite, sillä ohjelma ei ole kaatumaton. On hyvä tiedostaa myöskin että objekteja tallennettaessa ja ohjelman kaatuessa, on vaara että kyseiset tiedostot menetetään. Increment tallennuksilla vältetään katastrofeilta.

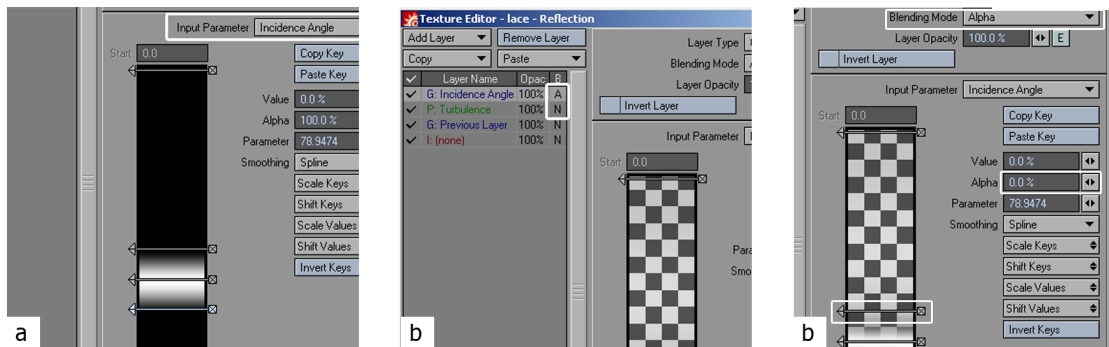
4.1 Layers

Surface editorin avatessa ilmestyy vasemmalta lista tekstuureista, jotka on listattu object kohtaisesti. Tekstuurit on hyvä nimetä jo alkuun loogisesti helpottamaan muokkauksia jälkikäteen. Oikealla puolella ikkunaa on varsinainen teksturointi osio, josta löytyy Basic palkin takaa tarpeelliset kanavat. Kun yksittäisen kanavan avaa ikkunaan, ikkunasta löytää listan layereita, joihin voi lisätä procedural tekstuurin, gradientin tai image mapin. Näitä layereita voi miksailla esimerkiksi alpha arvoin maskina toiselle layerille blending modea käyttäen kuten kuvassa 11 ja 12.



Kuva 11. Kuvassa a avattuna tekstuuri ikkuna. kuvassa b gradient Alpha arvolla

Alpha arvoa hyödyntäen on mahdollista esimerkiksi luoda tyylieltyjä valon heijastuksia incidence anglen mukaan. Tällöin kameran kulmasta riippuen esiintyy värejä. Tekstuuria voi käyttää esimerkiksi Luminosity, tai diffuse kanavaan jolloin laskenta on nopeaa.



Kuva 12. Kuvassa a input parameter; incidence angle. Kuvissa b; blending mode alpha

Alpha arvoja ja incidence anglea voisi hyödyntää mahdollisesti huijatus kuvan provisoimiseen pintaan, ja käyttää useampaa kuvaa suunnan mukaan. Mallinnuksissani päädyin käyttämään kuitenkin vain puhdasta valkoista. Layereitä en suosittelen käytettäväksi muuhun kuin nopeaan testaamiseen, jos silloinkaan sillä node alustan käyttäminen tekstuureissa on huomattavasti järkevämpää pitkällä tähtäimellä.



Kuva 13. Partahöylässä incidence anglen mukainen huijattu heijastus

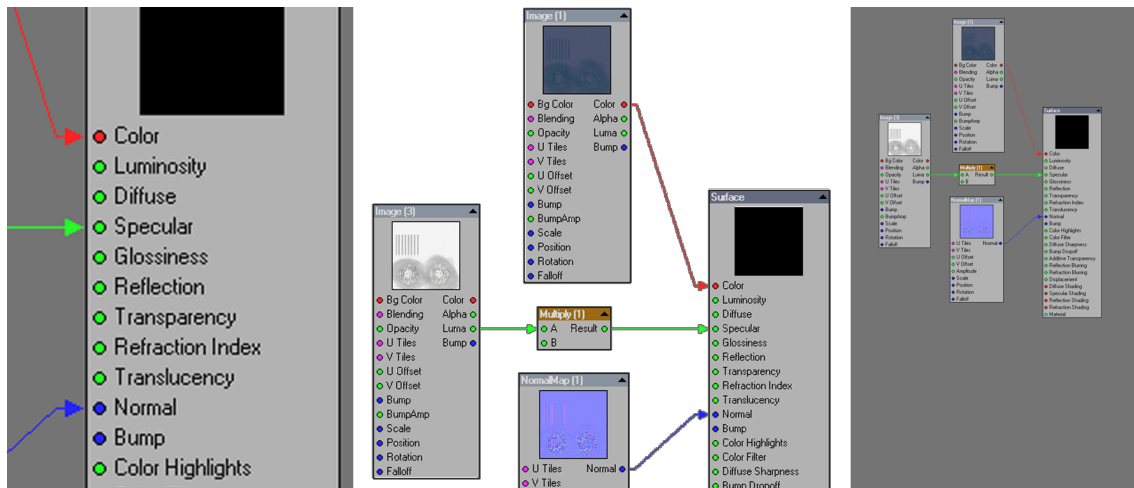
Todellisen heijastavan pinnan nopeaan testaamiseen voi laskea diffuse kanavan 50% ja kasvattaa reflectionin 50%. Heijastukset ovat laskennan kannalta kenties suurin hidastava tekijä. Jos heijastukset on mahdollista huijata visuaalisesti, tai käyttää muita tyyllillisiä keinoja, laskennan nopeuttamiseksi on se suositeltavaa. Environment paneelin alta käytetty reflection blurring varsinkin on suuri laskennan hidastaja, sillä sitä käytettäessä heijastuksen sample määrä on kasvatettava pehmentämään rakeita, ja näin renderoinnin säteiden laskentamäärä kasvaa, vaikka tuloksena onkin hienot pehmeät heijastukset.

Realistisuus ja oikeat heijastukset eivät välttämättä tyyllisestikään aina ole tavoiteltu

lopputulos.

4.2 nodes

Nodet mahdollistavat kokonaisuuksien nopean hahmottamisen, mikä on projekteissa olennaista mitä useampi tiimiläinen niitä jälkikäteen muokkaa. Node pohjaisuus vaatii ajattelua, mutta mahdollistaa monia tapoja, jotka normaalisti olisivat rajattuja. Koko rakenne on sekin aina näkyvissä ja nopeasti muokattavissa. Lightwave ohjelmassakin on tärkeää opetella node pohjaisuus, kun se on kerta tarjottu. Ainoa haitta node pohjaisuudesta on mahdollinen käyttäjän oman logiikan virheellisyys ja mahdolliset bugit ohjelmakoodissa. Seuraavassa kuvassa 14, esiintyy yksinkertainen node pohja jossa on tuotu Image map, specular ja color, surfacen inputteihin.

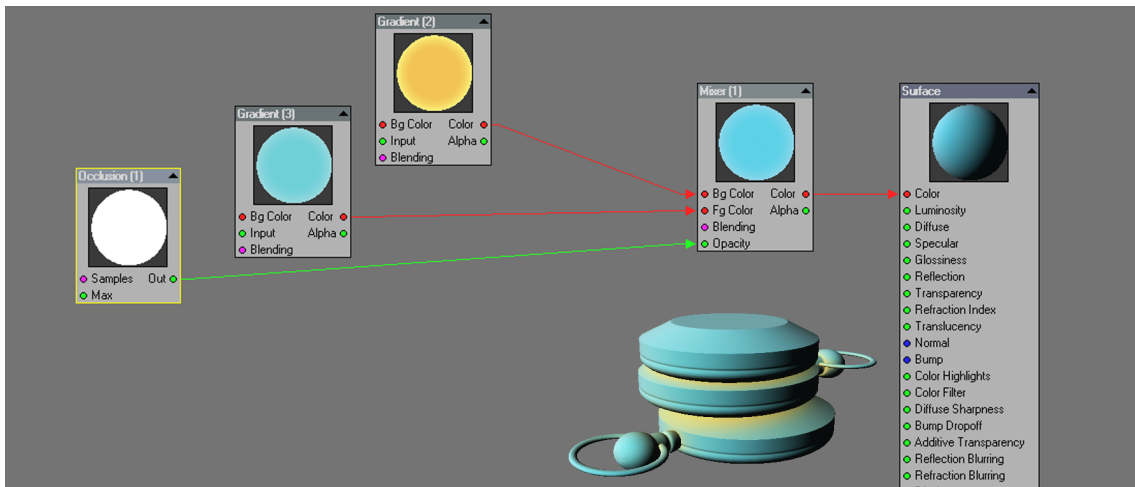


Kuva 14. Nodepuu

Image nodet on kertaalleen avattu ja valittu asetuksista oikeat uv mapit. Normal map on syötettynä erikseen. Specular map kanavassa on käytetty Luma output arvoa Alpha outputin sijaan. Näiden välillä ero on siinä, että Luma arvo katsoo kuvan kirkkausarvot, kun taas alpha etsii kuvan alpha kanavaa, jota tässä tapauksessa ei ole tallennettu, sillä kuvan formaatti on yhdistetty tiff. Luma arvoa on vielä manuaalisesti kerrottu suuremmaksi Multiply nodella.

Node pohjaisuus mahdollistaa esimerkiksi vaikkapa mixer nodea hyödyntäen occlusion laskennan käyttämisen kahden tekstuurin sekoittamiseen. Tällöin tummiin varjoalueisiin projisoituu enemmän toista tekstuuria. Mietin tämän hyödyntämistä esimerkiksi kivien tekstuurin sulattamiseen ympäröivään maastoon, mutta en vienyt ajatusta sen

pidemmälle, vaan käytin erilaisia miksaus tapoja kuten weight mappia.



Kuva 15. Occlusion nodella kahden värin miksaus ja lopputulos

Occlusion noden käyttämiseen renderointi asetuksissa Radiosity asetuksien ei tarvitse olla edes päällä, sillä node ohittaa laskennan ja suorittaa sen objektiokohtaisesti. Sample määrä valitaan noden sisältä. On huomioitava että esimerkiksi tällainen miksaus ei ole kovin kevyt ratkaisu, ellei lopputulosta yhdistä image mapiksi malliin bake prosessilla. Kuvassa 15 on havainnoitu miksaus noden toiminta, jossa on etualan ja taka-alan input sekä Opacity kanava scalar arvoilla sekoittamiseen.

Scalar arvot on kaikki merkitty vihrein värikoodein. Color output ja input väri on punaisella. Vektorit, sininen värikoodi, ovat esimerkiksi Normal ja Bump mapeissa käytettäviä arvoja, joilla on aina x, y, ja z mukainen suunta. Niiden muokkaaminen vaatii jo matemaattista ajattelukykyä. Math valikon alta löytyy matemaattisia apuvälineitä, joista kenties yksinkertaisin käytetty scalar arvoissa on multiply.

Teksturoinnin lisäksi nodeja voi käyttää esimerkiksi mallien displacement mappina ja mahdollistaa esimerkiksi animoimisen proceduraalien avulla.

4.3 procedural maps

Bitmap kuvat ovat sidottuja kuvakokoon pikselimäärään, kun taas procedural tekstuurit ovat matemaattisia algoritmeja. Procedural mapit voidaan luoda joko 3 uloitteisina tai 2 uloitteisina algoritmeina. Ghost Townin myötä tutustuin paljon IFW2sen tarjoamiin procedural mappeihin. Niiden etuna on tarkat tekstuurit, mutta vaativat paljon

muokkaamista ja kokeilemista halutun lopputuloksen saavuttamiseen, koska proceduraalit ovat matemaattisia ja arvot muuttuvat sen mukaan mitä vipuja proceduraalin kirjoittaja on käyttäjälle antanut. Ne myöskin herkästi antavat tietyn tyylin malleihin ja jos tahtoo enemmän kontrollia varsinkin alueittain tekstuureihin, kannattaa harkita image mappia.

Procedural map ei vie levytilaa kuin koodin verran ja yksittäinen proceduraali on näytönohjaimen muistin kannalta pienempi käsiteltävä kuin image map. Jälkikäteen procedural map viritelmää on vaikeampi lähestyä kuin olemassa olevaa tekstuuri tiedostoa, jonka voi renderoimatta tarkistaa. Procedural mapin kaikkein ongelmallisin puoli on tällä hetkellä se, että se saattaa aiheuttaa suuria antialiasing artefakteja ja myöskin proceduraaleista riippuen kohinan satunnaisuus voi ilmetä kuvasarjoissa (Yaldex). Koska kyse on koodista, on aina vaara ongelmiin, siinä missä image map on varma ratkaisu.

Procedural mapilla voi myöskin luoda tekstuuripohjan, jota käyttää uvmapin pohjalta bake toiminnoin ulos ajettuna photoshopissa, sekä syöttää image mappina jälleen tahdottuun kanavaan kuten esimerkiksi color tai diffuse inputtiin. Procedural algoritmeja voi käyttää myös scalar sekoittajina node puussa. Niiden käyttö on järkevää mutta on myös oltava varovainen, sillä lopputulos voi olla joskus yllättävä.

Valmiiksi kirjoitettuja procedural mappeja voi ostaa, kuten esimerkiksi IFW2 nodet lightwaveen, tai ne voi kirjoittaa itse jos tutkii aiheita enemmän. Niiden kirjoittaminen vaatii tosin koodaustaitoja ja matematiikkaa. Jo muutamalla procedural mapilla voi luoda suuren määrän erilaisia sekoituksia node pohjassa hyödyntäen scalar matematiikkaa. Proceduraleja voi tekstuurien ohella myös käyttää displacement mappiin ja erilaisiin animoinnin automatisoimisiin kuten esimerkiksi part move, joihin voi displacement node pohjassa hyödyntää proceduraalien kolmiulotteisia vektoreita.

5. Optimointi

Isoissa tiloissa, varsinkin ulkoympäristöissä muistinkäyttö voi kasvaa yli koneen suorituskyvyn, koska polygonimäärätkin helposti ovat suuria, ja jo pelkästään itse scenen lataaminen riippuu koneen muistimäärästä. Vaikkakin koneet ovat kehityksen ansiosta jatkuvasti tehokkaampia laskentaan, on hyvä huomioida vähemmän tehoa

sisältävien farmikoneiden määrä, suhteutettuna laskenta-aikaan jota ne voisivat lyhentää. Farmikoneet ovat tietokoneita, jotka laskevat animaation yksittäisiä kuvia verkon välityksellä. Toimiva farmi on edellytys raskaiden animaatioiden laskemiselle.

Renderointiajat kasvavat ilman hyvin suunniteltua rakennetta ja asetuksia. Tietokoneen laskenta-ajan nopeuttamiseksi on tärkeää optimoida aina kun mahdollista. Optimoinnilla tarkoitetaan laskenta-ajan vähentämistä kaikin mahdollisin keinoin, saavuttaakseen halutun tuloksen laadussa kärsimättä. Optimointia voi olla mm. renderointiasetusten ja renderoijan valinta, polygonimäärien vähentäminen, teksturoinnin optimointi, toimintojen varaaminen jälkikäsitteilyyn.

5.1 Polygonimäärän optimointi

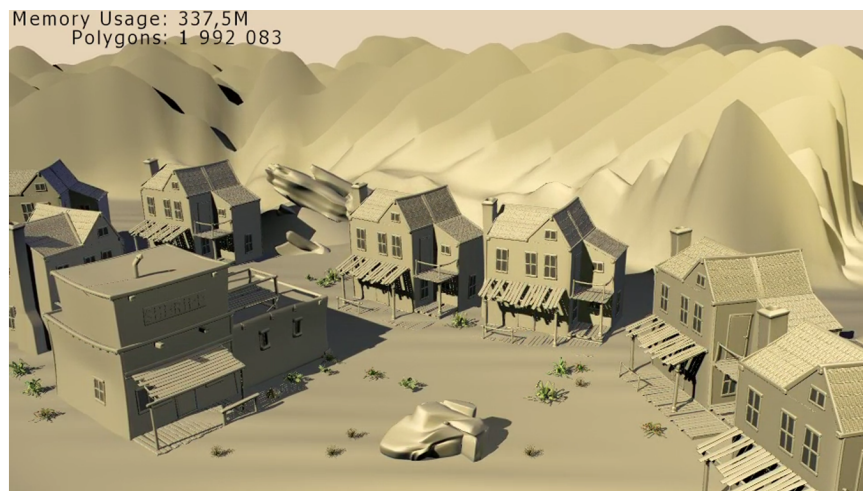
Polygonimäärä on täysin riippuvainen tekniikasta niin ohjelmisto tasolla kuin tietokoneen ja farmin tehoista. Käyttöjärjestelmän, ja ohjelmien on syytä olla 64-bit suuremman muistimäärän käyttöönottoon, sillä 32-bit fyysinen muistiraja tulee vastaan jo 4 GB (microsoft).

Lähtökohtana on se että malleissa ei ole mitään ylimääräistä mitä katsoja ei näe, ja että polygonimäärä on rajattu siihen mikä on tarpeen, eli polygonit itsessään eivät paista esiin. Polygonit ovat usein mallintajalle itselleen ja peliympäristössä hyväksyttävä tekijä, mutta viimeistellyssä animaatioissa katsojalle ei ole syytä esitellä karuja muotoja.

Käyttäessä subdivattuja malleja polygonimäärä moninkertaistuu ja samalla sillä on vaikutusta radiosityn laskentaan. Jotta malleja ei tarvitsisi tarkentaa enempää kuin tarpeen, voi niihin manuaalisesti lisätä geometriaa reunakohtiin ja olla käyttämättä subdivausta tai ainakin välttää suuret kertoimet. Mallista on syytä poistaa katsojalle ja laskennalle näkymättömät polygonit.

Subdivaus arvoja voi myös käyttää näkyvyyden mukaan shotti eli kuvakohtaisesti, rajata alueita pois laskennasta, tai tutkia mikä renderoinnissa vie eniten laskenta-aikaa. Malleihin voi käyttää normal mappauksia jos mahdollista, mikä helpottaa erityisesti radiosity laskentaa. Mikrodetailien muuttaminen tekstuuripohjaiseksi on kenties suurin polygonimäärää pudottavista tekijöistä. Detailit saattavat nostaa isoissa ympäristöissä muistinkäytön yli laskenta koneiden suorituskyvyn. Toisaalta myöskin suuret image

mapit vievät muistia ja suuri määrä normal mappeja voi kasvattaa muistinkäytön liian korkealle.



KUVA 16. Tekninen esimerkkikuva jossa polygonimäärä lähestyy jo kahta miljoonaa. Talot Tuomas Kankola, Leo Lipasti, Undo.

Mallintaessa yksittäinen optimoitu polygoni vastaa useampaa polygonia lopullisessa scene tiedostossa, polygonimäärien kertautuessa. Polygonimäärälle on hyvä antaa tavoitelukuja ja kertoa määrä sillä määrällä malleja kuin lopullisessa ympäristössä tulee olemaan. Tällöin voi hahmottaa onko detailitaso kaikkeen muuhun suhteutettuna järkevää vai onko syytä jopa karsia jostakin.

5.2 Mallit ja radiosity

Jos lasketaan radiosityn säteet kuviin, mallien polygonimäärästä riippuen radiosityn sädemäärä kasvaa. Except RadiosityGuide 95 oppaan mukaisesti radiosity laskee sampleja geometrian määrän mukaan eli lisää laskenta pisteitä. Näistä sampleista radiosity ampuu säteitä rays per evaluation arvojen mukaisesti (Except 2008). Radiosityn laskeminen monimutkaiseen malliin kasvattaa radiosityn laskenta aikaa. Vaikka itse radiosityn asetukset ja renderointi ei kuuluisi omaan toimeenkuvaan projektissa, on syytä huomoida se mallintaessa.

Joskus malleissa, kuten eräässä toisessa projektissa, tuli jopa vastaan ongelma, jossa ikkunan säleverhon geometria oli sellainen, että se päästi säteitä läpi osittain ja osittain ei. Kulmasta riippuen osa säteistä jäi kiinni ja osa jatkoi läpi arpomalla. Tällöin lopputuloksena oli jatkuvasti väreilevä valaistus. Ratkaisuna oli pois lukea sädeverho

laskennasta, mutta valaistuksen rakeisuuden syy oli silti pienen etsinnän alla. Mittasuhteet nähtynä säteisiin on siis otettava huomioon.

Jos myöskin käyttää tekstuureissa heijastuksia, on syytä aina miettiä minne säteet kimpoavat. Jos säteillä on mahdollisuus kimmota useammasta heijastuksesta ja vielä takaisin, on se jo suuri hidaste laskennassa. Renderoinnin optimointi on laaja alue, mutta polygonimäärällä voi aina vaikuttaa laskenta-aikaan hyvin paljon.

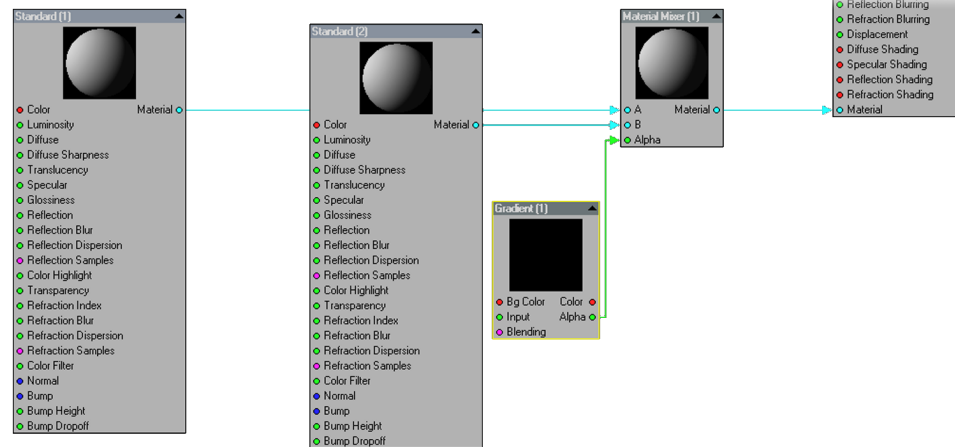
5.3 Tekstuurien optimointi

Lightwave tekstuuri node puun rakentaminen voi helposti karata käsistä, lopputuloksena hienolta näyttävä, mutta renderoinnissa raskas tekstuuri. Joskus rakennetun node puun, voi leipoa, englanniksi bake, moneksi tahdotuksi eri mapiksi, kuten diffuse, specular, shader, normal mapiksi, jotka sitten syöttää image mappina vastaaviin arvoihin, korvaamaan alkuperäisen tekstuuripuun. Optimoinnin kannalta tärkeä huomiokohde on node puussa materiaalien miksaus.

5.3.1 Materiaalit

Materiaali voi olla esimerkiksi carpaint, joka matkii maalipinnan lakkausta. Se on valmiiksi kirjoitettu pohja, jonka kaltaisen viritelmän voisi rakentaa myöskin node pohjassa yhdistellen kaikki matemaattiset toiminnot ja kanavat lopulta standard materiaaliin. Materiaali on siis node puussa eri laskutoimitusten ja kanavien yhdistetty yksittäinen taso. Kuten kuvasta 17 näkee, standard materiaalit sisältävät suuren määrän kanavia ja materiaaleja on myös tarjolla lukuisia erikseen kirjoitettuja, joita voi vielä miksailla monen toisen materiaalin kanssa, erinäisin miksaus tavoin.

Miksailulla saa hyvältä näyttäviä lopputuloksia, mutta joka kerta, kun materiaaleja miksataan, tai kun node puussa käytetään proceduraaleja, kone laskee jokaisen materiaalin nodepuun, niiden esiintyessä samaan aikaan. Miksatessa siis kahta eri materiaalia, lopputuloksena molemmat materiaalit siirtyvät laskentaan ja kaikki niiden käyttämät kanavat.



Kuva 17. Kaksi Standard materiaalia ja material mixer

Tutkiessani raskaan nodepuun materiaalin häivyttämistä kevyempään nodepuun materiaaliin etäisyyden mukaan kamerasta, minulle todettiin että tietenkin häivyttäessä kevyttä ja raskasta materiaalia, häivyttämisen keskialueilla molemmat materiaalit siirtyvät laskentaan, jolloin tällä välillä tekstuurin laskeminen on siis vielä raskaampaa kuin jo raskas materiaali. Näin ollen kyseinen materiaali optimointi ei ole mahdollista.

Ottaen huomioon myös kallioiden esiintyvyyden lähinnä tausta ympäristössä, monen eri proceduraalin sekoittaminen on turhaa ja virheille altista. Tästä syystä järkevintä oli käyttää bake menetelmää optimoimaan käytettyjä tekstuureja, tässä tapauksessa raskaan materiaalin ja yhdistelemään eri tasot kanavakohtaisesti.

5.3.2 Surface baking

Lightwave sisältää bake shaderin sekä bake kameran versiosta 9.0 ylöspäin, joista kamera on käytetympi, syystä että vanhempi shader on hitaampi, puutteita asetuksissa ja aiheuttanut myös forum lähteiden mukaan käyttäjillä ohjelman kaatuilua.

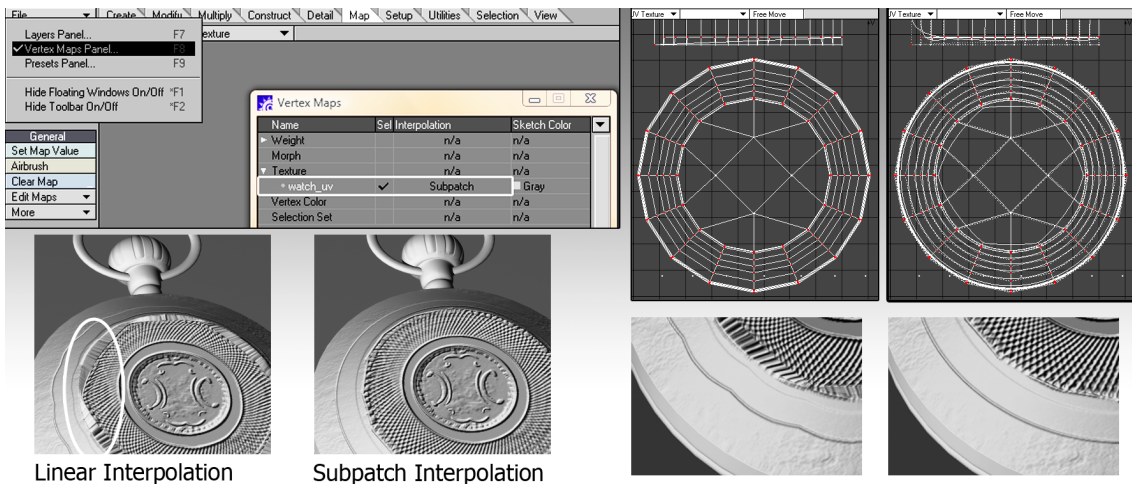
Bake kameran käyttö on suoraviivaista, mutta edellyttää uv mapin, joka lightwaven vielä puuttellisista uv työkaluista johtuen on suositeltavaa unwrapata jollakin toisella vaihtoehtoisella ohjelmalla. Oma suositukseni on modo, jolla on mahdollista tehdä molempia. Modo ohjelmassa tekstuurin leipominen high poly objektista low poly objektiin, toimii melko suoraviivaisesti bake from object toiminnolla high poly objektin ollessa tausta layerinä. Kuten monissa muissakin ominaisuuksissa, vaihtoehtoisia työtapoja löytää muista plugineista lightwaveen ja vaihtoehtoisista ohjelmista.

Diffuse mappiin ei ole syytä renderoida varjoja, jos se ei ole tarkoituksenmukaista. Suositeltavaa on käyttää subdivattua mallia, jotta tekstuuri ei veny. Uv mapit eivät saa myöskään olla päällekkäin, sillä se aiheuttaa mustia polygonialueita kuvaan. Mustiin kohtiin voi olla myöskin syynä bake kameran etäisyys asetuksissa.

5.2.3 Image maps

Image map on syytä laadullisesti pitää koossa, jossa yksittäiset pixelit eivät näy. Kuvasuhteiden muuttuessa esimerkiksi HD laatuun tekstuurien kokoa on pixelien näkyvyyden kannalta huomattavasti kasvatettava. Optimoinnin kannalta proceduraalit voittavat tässä niiden kestäessä useimmiten kuvakoon muutoksen.

Image mapin kanssa voi herkästi kompastua subdivausta käytettäessä, sillä kun polygonimäärää lisätään, ohjelma suorittaa polygonien lisäämisen normaalien mukaan, pehmentäen muotoja ja näin myös venyttäen ja kiristäen polygoneja, jolloin alkuperäiset uv-mapin etäisyydet, ja polygonimäärät ovat muuttuneet.



Kuva 18. Taskukellon uv interpolation korjaus

Tällöin on syytä unwrapata suoraan subdivide geometrialla, jonka voi lightwavessa kytkeä päälle mappia luodessa asetuksissa, tai vaihtaa sen jälkikäteen vertex map panelista, jossa Texture Interpolation Linear vaihdetaan Subpatch arvoksi, kuten kuvassa 18 tapahtuu.

Epämääräisen muotoiset polygonit aiheuttavat myöskin image mapin toistamisessa

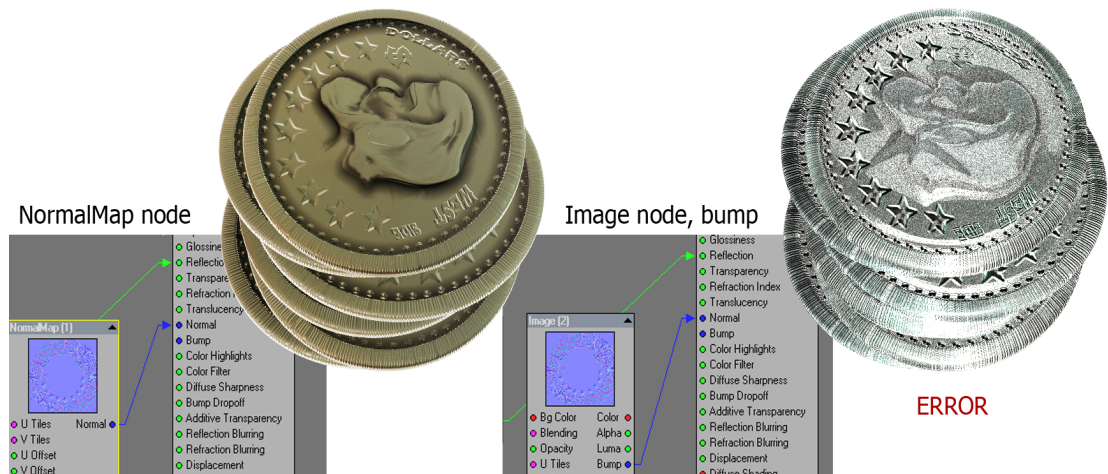
usein rikkinäisiä lopputuloksia. Hyvä polyflow ja tasaiset yksittäiset polygonit ovat Image mapin käyttämisen kannalta olennaista. Polygonin ollessa tasainen sen kohdalla toistettu tekstuuri toistuu varmemmin, kun taas yksittäisen polygonin vertexin venyessä suhteessa muihin sivuihin, Image map ei välttämättä toistu enää oikein näissä polygoneissa vaikka alueet olisikin unwrapatty hyvin (newtek forums 2009). On hyvä mallintaa pitäen polygonit pääasiassa suorakulmion muotoisina.

Normal map kuvien käyttäminen malleissa on tehokas keino säästää aikaa geometrian laskennassa. Normal map kuvat koostuvat 4 suunnasta RGB väriskaalan mukaan valaistusta kuvista.

Photoshop nvidia normal map filter mahdollistaa normal mapin konvertoinnin kuvasta. Normal map filterin tukema DDS tallennusmuoto on käytetympi peliympäristössä ja lightwave ei esimerkiksi muotoa tällä hetkellä tue. Kuvat voi kuitenkin tallentaa ulos esimerkiksi png tai tiff muodossa pakkaamattomana. Yksinkertaisissa tekstuureissa riittää tämä photoshop menettely ja graafiset taidot. Yksi vaihtoehtoinen ja nopea ohjelma kuvien tekemiseen olisi crazybump tutoriaaleista päätellen.

Jos on järjestelmäkamera käytössä voi myöskin tehdä tekstuurin kuvaamalla oikean kohteen yhdellä valolla valaistuna 4 eri suunnasta ja esimerkiksi photoshopissa säätää niiden sävyt tasoittain vastaamaan normal mapin RGB arvoja. Ryan Clark on esimerkiksi tehnyt tutoriaalin aiheesta (Clark). Tämä on kaikkein hitain ja epävarmin menetelmä, mutta on mahdollista jos siitä näkee itselleen hyötyjä. Normal mapin voisi vaikkapa innovatiivisesti valokuvata muovailuvahasta.

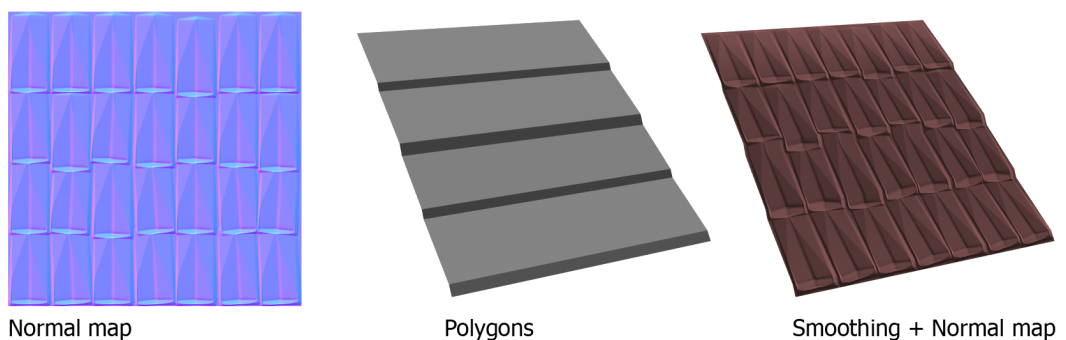
Lightwavessa Image map bump arvon kautta syötettynä kuvat eivät toimi normal inputissa vaan esiintyvät rakeisina, vaikka värikoodeina bump ja normal näkyvätkin molemmat vektoreina. Ohjelma ei siis osaa kääntää bump arvosta otettua kanavaa automaattisesti normaalin mukaan lasketuiksi vektoreiksi. On siis aina käytettävä tähän tarkoitettua NormalMap nodea. Myöskin esimerkiksi 3dsmax ohjelmassa normal map oli valittava erikseen ja syötettävä oikeaan kanavaan. Jos normal map ei ole syötetty oikein malliin, lopputulos voi näyttää kuvan 19 mukainen.



Kuva 19. Normalmap ja imagemap.

Ghost Townissa korjasin jälkikäteen työstämieni talojen katot yksityiskohtaisemmiksi ja karun proceduraalin sijaan mallinsin manuaalisesti normal mapin. Talon kattojen ollen erimuotoisia sekä myöskin nopean testauksen kannalta, näin järkeväksi mallintaa yksittäisen unwrapatyn palasen, joita voi sitten monistaa, liittää ja yhdistää jälkikäteen isommaksi kokonaisuudeksi. Jos kattoja katsottaisiin lähietäisyydeltä, kyseinen normal map ei olisi enää sopiva. Tein normal mapin myös pieneen kokoon ja nopean kaavan kautta suhteutettuna polygonien vähäiseen määrään.

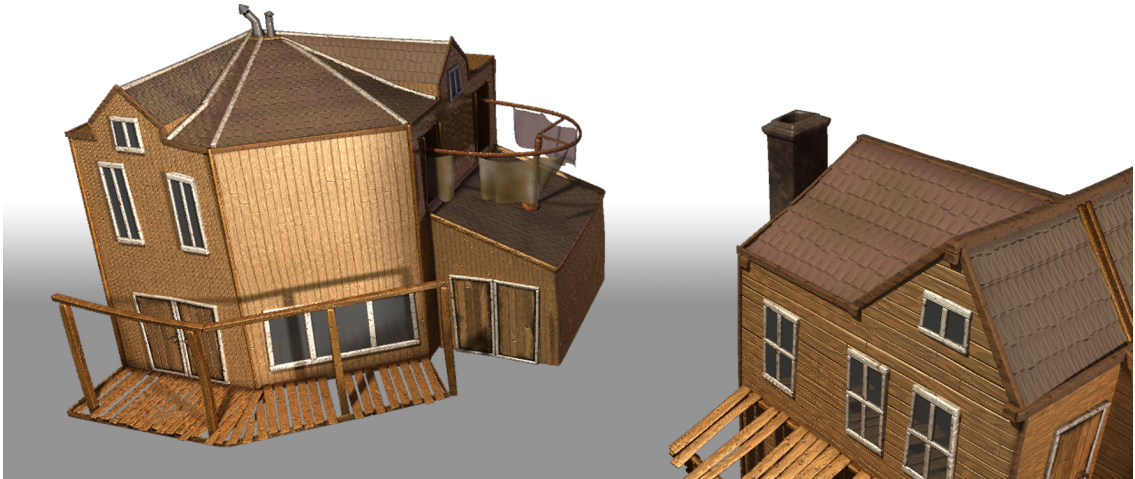
Mallinsin yhden high poly katto tiilen, jota monistin tekstuurin alueelle sisäkkäin ja vierekkäin kuten kattotiilet olisivat. En nähnyt tarpeelliseksi odottaa normal map renderointiä vaan renderoin tiilet pelkällä slope gradientilla tummasta valkoiseen y-akselin mukaisesti, ja vieden kuvan suoraan photoshoppiin normal map filterointiin, jonka lopputulos kuvassa 20.



Kuva 20. Normal map

Yksittäisen osan mallintamisen ja monistamisen vahva puoli on siinä, että uv säilyy kauttaaltaan ja on muokattavissa matkan varrella. Tällöin tekstuuri monistuu tile

ominaiseen tapaan ja hyvin kontrolloidusti. Jokaisen katon unwrappays ja teksturointi olisi ollut toinen lähestymistapa, mutta tällöin jokaisella katolla olisi oma tekstuuri ja kaikkien kattojen tekstuuria ei voisi vaihtaa yhtä kuvaa vaihtamalla, kuten tässä tapauksessa voi tehdä. Katoissa olisi mahdollista käyttää myös cube mapping projisointia, mutta sen säädöt on tehtävä myöskin katto ja mallikohtaisesti, eikä ajallisesti ole kovinkaan paljon nopeampi prosessi.



Kuva 21. Talot normal map katoilla. Mallit Tuomas Kankola, Leo Lipasti, Undo.

Kyseinen katto on optimoinnin kannalta hyvin kevyt. Procedural tekstuurin käyttäminen katoissa on nopeaa, mutta koska tässä tapauksessa katon normal map on helppo vaihtaa jälkikäteen yhtä kuvaa vaihtamalla ja koska proceduraalien määrä on hyvä pitää rajatuissa määrissä, oli tämä järkevä vaihtoehto. Kaikkien kylän kattojen vaihtaminen tällä tapaa on tosin melko hidas prosessi. Procedural tekstuurien, image mappien ja sitä myöten unwrappauksen tarpeellisuus on tasapainottelua. Image mapit toimivat varmasti, kun taas Lightwavessakin on esiintynyt joissain kuvasarjoissa farmin laskennan ja procedural tekstuurien yhdistelmänä kohinaa johtuen satunnaisista arvoista, jotka liikkuvat matemaattisissa funktioissa. Joskus tietyt ominaisuudet sisältävät ohjelmakohtaisesti bugeja ja ne on vain huomattava tarpeeksi ajoissa.

6. Loppupohdinta

Usein joko ennalta mietittynä, tai itse mallinnusprosessin aikana huomaa, mikä tekee tietystä esineestä, tai asiasta sen oman olemuksen, piirteet ja käyttötarkoituksen. Luonnossa yhteyttävät kasvit ja puut kasvavat kohti valoa. Niiden hidasta kasvua estää vain fysiikan lait ja ilmasto, jonka takia ne kasvavat miten kasvavat. Tämä mielikuva yhdistettynä olemassa oleviin referensseihin mahdollistaa vaikka täysin uuden kasvin

luomisen. Luonnonilmiöiden, mekaniikan, fysiikan ja pinnallinen tietämys auttaa tekemään perusteltuja ja samalla luovia ratkaisuja.

Luovuuden ylläpitämiseksi väitän että mallintajan tärkeimpiä lehtiä ovat tiedelehdet. Inspiraatiota tukevat myös omakohtaiset kokemukset ja käytännön tieto. Esimerkiksi kukapa osaisi luoda kattauksia paremmin kuin mallintaja joka on joskus toiminut tarjoilijana. Referenssit ovat yleisempiä mallien lähteitä, mutta mielikuvituksellisen ja funktionaalisen, eli toinen malli toisen olemusta ja tarkoitusperää yhdessä tukien, ja yhtenäisen ympäristön luomisen kannalta, on tärkeää ymmärtää mitkä ovat ympäristössä yhteisiä tekijöitä, kuten esimerkiksi eroosio.

Katsojalle on järjellisesti yhdentekevää miten ympäristö on muovautunut, koska pääosin ajatuksella hän seuraa animointia ja juonta, mutta väitän, että tyylielityksessä maisemassakin alitajuisesti katsojaa kiehtoo pienet nyanssit, yksityiskohdat ja realismia lähestyvät yksittäiset asetelmat. Luonto itsessään, joka on ihmistä isompi ja monesti kummallinen, toimii mallinnettuna jo itsessään esimerkiksi tunnelmaa nostattavina välikuvina, sekä introissa, jota on käytetty tehokeinona sarjoissa kautta aikain. Yksi hyvä esimerkki tästä on Twin peaksin pilotti (kuva 22), jossa jo introssa ympäristöä kuvaamalla luotiin koko käsitys sarjan tapahtumapaikasta sekä samalla romantiikan tyyliin tapaan, tuntemuksien herättämistä katsojassa maiseman ja kohteiden avulla. Ympäristön mallinnusta ei kannata siis väheksyä, ja jättää puolitiehen vaan se on syytä rakentaa ajatuksella, niin visuaalisesti kuin teknisestikin, sillä se puhuu katsojalle.



Kuva 22. Twin Peaks intro.

Mallintaminen itsessään on väline. Väline jolla saavutetaan oikea pohja visuaaliselle tuotteelle. Ghost Town mallinnustyö jatkuu tätä kirjoittaessani. Aika on määre josta on aina pulaa. Joskus tosin riittää hyvä pohja, jota voi työstää iteraatioin eteenpäin.

Lähteet:

Evans, Christopher 2003. An introduction to modeling with subpatch weight in Lightwave 3D [verkkodokumentti]

<http://chrisevans3d.com/tutorials/weights.htm> (luettu 06.12.2009)

Except 2008. Radiosity opas [verkkodokumentti].

<http://www.except.nl/lightwave/RadiosityGuide95/> (luettu 06.01.2010)

Clark, Ryan. Normal map photography tutoriaali [verkkodokumentti].

<http://zarria.net/nrmphoto/nrmphoto.html> (luettu 30.04.2010)

Yaldex.com. Chapter 11. Procedural texture shaders [verkkodokumentti].

<http://www.yaldex.com/open-gl/ch11.html> (luettu 14.02.2010)

Newtek forums 2009. Surface baking in lightwave [verkkodokumentti].

<http://www.newtek.com/forums/showthread.php?t=95689> (luettu 27.04.2010)

Newtek forums 2009. 3D Painting: Seamless Rendering UV maps for Subdivision Models? [verkkodokumentti].

(luettu 08.03.2010)

Newtek forums 2008. Catmull-clark vs subpatch [verkkodokumentti].

<http://www.newtek.com/forums/showthread.php?t=91533> (luettu 08.03.2010)

Microsoft. Memory Limits for Windows Releases [verkkodokumentti].

<http://msdn.microsoft.com/en-us/library/aa366778%28VS.85%29.aspx> (luettu 05.06.2010)

Blevins. Interview: Inside Pixar with Neil Blevins [verkkodokumentti]

<http://www.thunderchunky.co.uk/articles/inside-pixar-with-neil-blevins/> (luettu 05.06.2010)



Ghost Town

Viestinnän koulutusohjelma
3D-visualisointi
Opinnäytetyö
7.6.2010

Leo Lipasti

Degree Programme in Media		Specialisation 3 D Visualisation
Author Leo Lipasti		
Title A modelling process of a project		
Tutor(s) Jaro Lehtonen		
Type of Work Bachelor 's thesis	Date June 2010	Number of pages + appendices 30
<p>The aim of the Thesis was to look into the concept of general creativity as a key component necessary in a modelling process, starting points, and in modelling and optimization methods. Ghost Town, an animated series, executed by Undo Ltd, provided the basis and primary material for the examination.</p> <p>Essential methods of working with polygons and ways of enabling rapid modelling were examined in detail, which offered various approaches, also planning and preparations to achieve an effective work process. Lightware programme was utilized to illustrate how to use some of its specific features.</p> <p>The Thesis will be useful to a reader who has not yet had experience, or has little experience in the modelling process of a project.</p>		
Work / Performance / Project		
Place of Storage Helsinki Metropolia University of Applied Sciences, Tikkurila Campus Library		
Keywords Modeling, animation, Ghost Town		

TIIVISTELMÄSIVU

Koulutusohjelma Viestintä		Suuntautumisvaihtoehto 3D-visualisointi	
Tekijä Leo Lipasti			
Työn nimi Ghost Town			
Työn ohjaaja/ohjaajat Jaro Lehtonen			
Työn laji Opinnäytetyö		Aika 7.6.2010	Numeroidut sivut + liitteiden sivut 30
<p>TIIVISTELMÄ</p> <p>Opinnäytetyö koostui yleisestä mallintamiseen tarvittavasta luovuudesta, lähtökohdista, mallinnustavoista sekä mallinnusten optimoinnista. Työn pohjana käytettiin työstön alla olevaa Ghost Town animaationsarjaa, jonka toteuttajana toimii Undo Oy:n tiimi.</p> <p>Työssä käytiin läpi esimerkein polygonimallinnukselle olennaisia työskentelytapoja ja pikamallinnukseen mahdollistavia keinoja, tarjoten ajatusmalleja ja valmisteluja tehokkaan työskentelyn saavuttamiseksi.</p> <p>Käsittelin Lightwave ohjelmaa ja kuinka hyödyntää sen tarjoamia ominaisuuksia valituin esimerkein. Opinnäytetyöstä on hyötyä lukijalle, jolla ei vielä ole harjaannusta mallintamisen prosessista projektissa.</p>			
Teos/Esitys/Produktio			
Säilytyspaikka Metropolian kirjasto, Tikkurilan toimipiste			
Avainsanat Mallintaminen, animaatio, Ghost Town			